1. A processor comprising:

first and second instruction pointer (IP) sources to provide IPs for first and second instruction threads, respectively;

an instruction cache to provide a cache line responsive to an IP;

a source arbiter to provide an IP from the first or second IP source to the instruction cache;

an instruction buffer (IB) to receive a first block of the cache line; and

a temporary instruction cache (TIC) to receive a second block of the cache line.

2. The processor of claim 1, wherein the IB and the TIC receive the first and second blocks of the cache line on a first clock interval.

3. The system of claim 2, wherein the second block of the cache line is transferred to the IB on a subsequent clock interval.

4. The system of claim 3, wherein the source arbiter provides IPs from the first and second sources on alternate clock intervals.

5. The system of claim 4, wherein the IB includes first and second IBs to store instructions from the first and second threads, respectively.

6. The system of claim 5, wherein the first IB receives an instruction block from the TIC and the second IB receives an instruction block from the cache on a second clock interval.

7. The processor of claim 1, wherein the IB receives first and second instruction blocks from the instruction cache and the TIB, respectively, on adjacent clock intervals.

8. The processor of claim 1, wherein the IB comprises first and second IBs and the instruction cache provides instruction blocks to the first and second IBs on adjacent clock cycles.

9. The processor of claim 8, wherein the TIC provides instruction blocks to the first and second IBs on adjacent clock cycles.

10. An instruction fetch engine comprising:

an instruction cache to provide a line of instructions in response to an instruction pointer;

an instruction queue to receive a first block of the instruction line during a first clock interval; and

a temporary instruction cache to receive a second block of the instruction line during the first clock interval.

11. The instruction fetch engine of claim 10, wherein the temporary instruction cache provides the second block of the instruction line to the instruction queue during a subsequent clock interval.

12. The instruction fetch engine of claim 10, wherein the instruction cache stores lines of instructions for first and second instruction threads and the instruction queue includes first and second instruction queues to store blocks of instructions for the first and second instruction threads, respectively.

13. The instruction fetch engine of claim 12, wherein the instruction cache provides first and second blocks of a line of instructions for the first instruction thread to the first instruction queue and the temporary instruction cache, respectively, during the first clock interval.

14. The instruction fetch engine of claim 13, wherein the instruction cache provides first and second blocks of a line of instructions for the second instruction thread to the second instruction queue and the temporary instruction cache, respectively, during a second clock interval.

15. The instruction fetch engine of claim 10, wherein the instruction queue includes first and second instruction queues and the instruction cache provides first blocks of instruction lines for the first and second instruction threads to the first and second instruction queues on alternate clock intervals.

16. The instruction fetch engine of claim 15, wherein the instruction queue provides second blocks of the instruction lines for the first and second instruction threads to the temporary instruction cache on alternate clock intervals.

17. The instruction fetch engine of claim 16, wherein the temporary instruction cache provides the second blocks of the instruction lines for the first and second instruction threads to the first and second instruction queues, respectively, on alternate clock intervals.

18. A method comprising:

selecting first and second cache lines for first and second instruction threads;

providing first and second instruction blocks of the first cache line to a first instruction queue and a temporary instruction cache, respectively, during a first clock interval; and

providing first and second instruction blocks of the second cache line to a second instruction queue and the temporary instruction cache, respectively, during a second clock interval.

19. The method of claim 18, further comprising providing the second block of the first cache line to the first instruction queue during the second clock interval.

20. The method of claim 19, wherein selecting first and second cache lines comprises:

receiving instruction pointers for the first and second threads; and

providing first and second cache lines responsive to receipt of the instruction pointers for the first and second threads, respectively.

21. The method of claim 20, wherein receiving instruction pointers for the first and second threads comprises receiving instruction pointers for the first and second threads during adjacent clock intervals.

22. The method of claim 21, wherein providing the first and second instruction blocks comprises providing the first and second instruction blocks during adjacent clock intervals.

23. A system comprising:

1^{st} through n^{th} instruction pointer (IP) sources;

1^{st} through n^{th} instruction queues associated with the 1^{st} through n^{th} IP sources, respectively;

a cache to provide a first portion of an instruction block to one of the 1^{st} through 4^{th} instruction queues, responsive to an IP from the associated IP source; and

a temporary storage to receive a second portion of the instruction block.

24. The system of claim 23, wherein the block of instructions is a cache line that comprises n portions of instructions.

25. The system of claim 23, further comprising an arbiter to select an IP from one of the 1st through 4th IP sources to send to the cache.

26. The system of claim 23, wherein the cache provides the first portion of the instruction block to the one of the 1st through 4th instruction queues on a first clock interval and the temporary storage structure provides a second portion of the instruction block to the one of the 1st through 4th instruction queues on a second clock interval.